



Patent
Attorney's Docket No. 0220-087

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)	BOX Appeal Brief-Patents
Robert M. HANEVOLD)	
Application No.: 09/977,143)	Group Art Unit: 2178
Filed: October 12, 2001)	Examiner: Kyle Stork
For: METHOD FOR PREVENTING)	
INADVERTENT DATA ENTRY IN A)	
WEB PAGE)	

Commissioner for Patents
Alexandria, VA 22313-1450

APPEAL BRIEF PURSUANT TO 37 C.F.R § 41.37

Sir:

Further to the Notice of Appeal filed on April 9, 2007 in connection with the above-identified application submitted herewith is the Appeal Brief.

07/10/2007 RFEKADUI 00000019 09977143

01 FC:140E

500.00 0P

(i) **REAL PARTY IN INTEREST**

The real party in interest is the assignee, BellSouth Intellectual Property Corporation.

(ii) **RELATED APPEALS AND INTERFERENCES**

To the best of the undersigned's knowledge, there are no related appeals or interferences.

(iii) **STATUS OF CLAIMS**

Claims 1-18 and 24 are currently pending, have all been rejected two or more times, with the exception of claim 24 which has been rejected once, and are all the subject of this appeal. Claims 19-23 have been cancelled.

(iv) **STATUS OF AMENDMENTS**

No Amendments have been submitted in this application subsequent to the Final Office Action dated December 11, 2006.

(v) **SUMMARY OF CLAIMED SUBJECT MATTER**

The exemplary embodiments describe methods and systems for preventing unintended entry or submission of data via a Web browser or the like.

According to exemplary embodiments, independent claim 1 describes a method for preventing data entry via a data input screen on a client device, comprising: rendering source code that defines the data input screen in the client device (see, e.g., paragraph 033 of the application); defining an executable script within the source code (see, e.g., paragraph 033 of the application); and executing said executable script in response to user input, wherein said executable script operates within said client device to render said data input screen inaccessible to prevent subsequent user input (see, e.g., paragraphs 033-034 of the application); wherein said step of executing further comprises the steps of: associating said executable script with a predetermined z-index number for a web page; and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number (see, e.g., paragraphs 033-034 of the application).

According to exemplary embodiments, independent claim 5 describes an apparatus for preventing entries or submissions of data via an input screen displayed on a client device, comprising: a central processing unit (see, e.g., paragraph 028 and Figure 2 of the application); a memory (see, e.g., paragraph 028 and Figure 2 of the application); a user input device (see, e.g., paragraph 028 and Figure 2 of the application); a display (see, e.g., paragraph 028 and Figure 2 of the application); and a browser adapted to render said input screen on said display, wherein source code is provided to said browser that contains instructions that are interpreted by said browser to render said input screen inaccessible after an executable script contained within said source code is executed on said client device (see, e.g., paragraphs 033-034 of the application); wherein said source code further contains instructions which operate to: generate association of said executable script with a predetermined z-index number for a web page (see, e.g., paragraphs 033-034 of the application); and render inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number (see, e.g., paragraph 034 of the

application).

According to exemplary embodiments, independent claim 10 describes a computer-readable medium having computer-executable components comprising: a form definition component defining a data input screen and a data submission field (see, e.g., paragraph 033 of the application); a style definition component defining a layer having a width and height at least as large as said data submission field (see, e.g., paragraph 033 of the application); a function definition component responsive to said data submission field, wherein upon execution of said function definition component, said layer operates to render said data submission field inaccessible on said form (see, e.g., paragraphs 033-034 of the application); wherein said computer-executable components are operable to perform the steps of: associating said executable script with a predetermined z-index number for a web page, and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number (see, e.g., paragraphs 033-034 of the application).

According to exemplary embodiments, independent claim 15 describes a method for preventing data entry to a server computer from a client computer, comprising: receiving a request for an exchange of data from said client computer (see, e.g., paragraph 033 of the application); defining an executable script within a source code, said executable script operating in response to a client computer input and rendering a data input screen inaccessible to prevent subsequent input from said client computer (see, e.g., paragraphs 033-034 of the application); and providing said source code that defines said data input screen (see, e.g., paragraph 033 of the application); wherein said step of defining further comprises the steps of: associating said executable script with a predetermined z-index number for a web page; and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number (see, e.g., paragraph 034 of the application).

According to exemplary embodiments, independent claim 18 describes a method for preventing data entry to a web page comprising the steps of: associating an executable script with said web page (see, e.g., paragraph 033 of the application);

permitting a first data input to said web page (see, e.g., paragraph 033 of the application); executing, in response to said first data input, said executable script (see, e.g., paragraph 033 of the application); and preventing data entry to at least a portion of said web page after execution of said script, wherein said step of preventing further comprises the steps of: associating said executable script with a predetermined z-index number for said web page (see, e.g., paragraph 033-034 of the application); and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number (see, e.g., paragraphs 033-034 of the application).

According to exemplary embodiments, independent claim 24 describes a method for preventing data entry to a web page comprising the steps of: associating an executable script with said web page (see, e.g., paragraph 033 of the application); determining if said web page uses z-index numbers (see, e.g., paragraph 033 of the application); permitting a first data input to said web page (see, e.g., paragraph 033 of the application); executing, in response to said first data input, said executable script (see, e.g., paragraph 033 of the application); and preventing data entry to at least a portion of said web page after execution of said script, wherein said step of preventing further comprises the steps of: associating said executable script with a predetermined z-index number for said web page if said web page supports using said z-index number (see, e.g., paragraphs 033-034 of the application); associating said executable script with a division of said web page if said web page does not support using said z-index number (see, e.g., paragraphs 033-034 of the application); rendering inaccessible those data entry elements associated with said web page by rendering said division of said web page visible over said data entry elements if said web page does not support using said z-index number (see, e.g., paragraphs 033-034 of the application); and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number if said web page supports using said z-index number (see, e.g., paragraphs 033-034 of the application).

(vi) **GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

A number of grounds of rejection are raised by the Examiner and listed below. Appellant requests review of each of these grounds of rejection on appeal.

a. Claims 1-17 and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Brown et al. (U.S. Patent Number 6,278,448) and further in view of Barlow et al. (U.S. Patent Number 6,275,935) and further in view of Humes (U.S. Patent Number 6,539,430).

b. Claim 18 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Moneymaker et al. (U.S. Patent Application Number 2002/0049708) and further in view of Humes (U.S. Patent Number 6,539,430).

(vii) **ARGUMENT**

The Brown patent describes a method of creating a composite desktop built by a user from Web content retrieved from one or more Web sites. The Barlow patent describes a system for preventing unauthorized modification of interactive objects having one or more object states by an object designer. The Humes patent describes a system and method for filtering data received by a computer system. Multiple features claimed by Appellant are not reached by any combination of these three references, nor would there have been sufficient motivation to have combined these three references in such a manner as to reach the position of obviousness stated in the Final Official Action dated December 11, 2006. Specific examples with respect to the claims will be shown below.

a. Independent Claim 1

A method for preventing data entry via a data input screen on a client device, comprising:
rendering source code that defines said data input screen in said client device;
defining an executable script within said source code; and
executing said executable script in response to user input,
wherein said executable script operates within said client device to render said data input screen inaccessible to prevent subsequent user input;
wherein said step of executing further comprises the steps of:
 associating said executable script with a predetermined z-index number for a web page;
and
 rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number.

The Brown patent describes a method of creating a composite desktop built by a user from Web content retrieved from one or more Web sites. The composite desktop can contain objects which represent Uniform Resource Locators (URLs) to either static or dynamic content. Additionally, in a preferred embodiment within the Brown patent, a user can set up multiple desktops and have the ability to switch between them. Per the Final Official Action, Brown "fails to specifically disclose rendering the data input screen inaccessible to prevent user input", and attempts to use the Barlow patent to remedy this deficiency.

Barlow describes a system for preventing unauthorized modification of interactive objects, having one or more object states, by an object designer. Preventing unauthorized access occurs through a locking apparatus and allows an object designer to prevent end users from modifying one or more aspects of the object. Two examples of locking methods mentioned in Barlow are locking through the use of a password and locking the object based upon the location of display of the object in question.

Regarding independent claim 1, the Final Official Action also correctly states that "Brown further fails to disclose associating the executable script with a predetermined z-index number for a web page and rendering inaccessible those data entry elements associated with the web page that have a z-index number lower than the predetermined z-index number." The secondary patent to Barlow also fails to teach or suggest this feature of Appellant's claim 1 combination (among other things) as evidenced by the Final Office Action's reliance upon the tertiary patent to Humes.

The Humes patent describes a system and method for restricting access to data by filtering data, such as a Web page, received by a computer system. This filtering method works by checking to see if the URL has been pre-approved or pre-denied. Assuming that the URL has been pre-approved, the header of the web page is filtered, followed by the body of the web page. This allows portions of data to be seen by an end user based upon the filtering parameters.

However, Appellant respectfully disagrees that it would have been obvious to one of ordinary skill in the art at the time of the Appellant's invention to have combined the teachings of Brown with the teachings of Barlow and Humes in any manner which would have motivated one of ordinary skill in the art to have arrived at Appellant's claim 1 combination for at least the reasons stated below.

1. Even Assuming (Strictly Arguendo) That There Was Some Motivation to Combine Brown with Barlow and Humes, the Resulting Combination Would Not Have Resulted in Appellant's Claim 1 Combination.

As mentioned above, neither the primary reference to Brown nor the secondary

reference to Barlow teaches or suggests the last two steps of claim 1, namely:
“associating said executable script with a predetermined z-index number for a web page; and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number.” Thus the Final Official Action relies upon Humes. The cited sections of Humes (Figure 5; col. 2, lines 36-49), used in the Final Official Action to allegedly remedy the deficiency of Brown et al. and Barlow et al., describe a method of filtering a page based on weighted values of the words found on the page to be accessed. For example, part of the associated text description of Figure 5 (col. 9, lines 57-63) of Humes is shown below:

“Decision block 522 determines whether the ‘Score’ for the page or block of text exceeds the predetermined ‘Targetscore’ threshold and, if so, the page or block of text is replaced with the ‘FORBIDDEN’ page or message in block 526 before a ‘Yes’ is returned by terminal block 528, indicating that access was denied based on the web page body.”

This method of filtering used in Humes is not the same as “wherein said step of executing further comprises the steps of: associating said executable script with a predetermined z-index number for a web page; and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number” which is found, among other things, in Appellant’s claim 1 combination.

Initially it is noted that two text searches were performed upon the Humes patent as found on the USPTO website for the phrases “z-index” and “z-number”. Neither search of the Humes patent resulted in any hits. Only Appellant’s specification teaches or suggests this feature in combination with the other factors of Appellant’s claim 1 combination.

Since there is no reference to a z-index or z-number in the Humes patent, it appears that the Final Official Action incorrectly equates the Targetscore of Humes to the z-index used in Appellant’s claim 1 combination as shown by the following from the Final Official Action:

“Humes discloses associating the executable script with a predetermined z-index number for a web page and rendering inaccessible those data entry elements associated with the web page that have a z-index number lower than the predetermined z-index number (Figure 5).”

Figure 5 of Humes describes a filtering process. More specifically, therein if a score associated with objectionable words on a particular web page is less than the predetermined Targetscore, then the web page is shown with the objectionable words replaced. Alternatively, if the score of the objectionable words is greater than the Targetscore, then the web page is not shown.

By way of contrast, a z-index number is often used in cascading style sheets (CSS) for displaying hypertext markup language (HTML) objects or other similar programming areas that need to have a system for determining the overlap order of displayed objects, frames, windows, etc. For example, if two objects were to overlap each other on a two dimensional display, the object with the higher z-index number would be displayed in front of the object with the lower z-index number. A purely illustrative example of how the z-index can be used can be found in a Microsoft article entitled "INFO: How the z-index Attribute Works for HTML Elements", revision 3, dated May 11, 2006 and it can be located online at <http://support.microsoft.com/kb/177378>.

More importantly, even if one modified the combination of Brown and Barlow to include the filtering mechanism of Humes, the resulting combination would not reach claim 1 because:

- (a) the Targetscore of Humes is not a z-index number; and
- (b) the filtering mechanism of Humes does not "render inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number". Instead, the filtering mechanism of Humes either replaces objectionable words or prevents a web page from being shown in its entirety.

Yet another reason why this combination of references cannot possibly reach Appellant's claim 1 combination can be found earlier in the claim. More specifically, while the filtering of objectionable data of Humes prevents a user from seeing objectionable data, Appellant's claim 1 combination provides operates to "render said data input screen inaccessible to prevent subsequent user input" (emphasis added). Therefore, even assuming (strictly arguendo) that one of ordinary skill in the art were motivated to combine Brown et al., Barlow and Humes in the manner suggested, the

result would at best (regarding this claim feature) filter out incoming undesirable data, and not restrict access to data entry elements in the system of Brown et al.

2. There Would Have Been No Motivation to Have Combined Brown and Barlow, In the Manner Described.

Moreover, there would have been no reason why one of ordinary skill in the art would have combined Brown and Barlow in the manner described in the Official Action in the first instance.

In the Final Official Action it is stated that:

"It would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to have combined Brown's method with Barlow's method, since it would have allowed a user to restrict access to data (Barlow: column2, lines 8-10)."

It is respectfully submitted, however, that there is no basis found in either Brown et al. or Barlow et al. for the proposed motivation to combine because neither Brown et al. nor Barlow et al. suggests that there would be any desirability or need for allowing a user to restrict access to data in the particular system of Brown et al. For example, as stated in column 3, lines 35-48 of Brown:

"a system and method for creating a composite desktop from Web content provides a way of displaying composite desktop components on a desktop in a readily available manner. The invention allows a composite desktop to display images from any Web page in front of a back-ground wallpaper. The invention further allows the display of active components that regularly update their display according to preprogrammed mechanisms, such as retrieving new data from a Web site." (emphasis added)

The above section of Brown shows no desire or motivation to restrict access to data. If anything, it teaches away from restricting access to data. Thus the proposed rationale for combining Brown and Barlow to reach the initial combination in the chain which forms the basis for the obviousness rejection is respectfully submitted to be flawed.

3. There Would Have Been No Motivation to Have Modified the Combination of Brown and Barlow Based on Humes.

Still further, there would have been no motivation for one of ordinary skill in the art to have continued the chain of modifications to then have modified the combined system of Brown and Barlow as described in the Official Action, based on Humes. Specifically, the Official Action states:

"Further, it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to have combined Humes with Brown, since it would have allowed a user to filter objectionable data (Humes: column 2, lines 36-49)."

However, as mentioned above, Brown is interested in broadly providing a composite desktop to display images from any Web page in front of a back-ground wallpaper. Thus there would have been no motivation to have provided the automated filtering technique of Humes to the combination of Brown and Barlow for the same reasons given above, i.e., this rationale presumes a desirability for censorship which is not evident from the teachings of the references.

Claims 2-17

It is respectfully submitted that these claims are allowable for at least the reasons set forth above with respect to claim 1. Additionally the dependent claims are allowable for reasons of their own. For example, dependent claim 9 describes "the apparatus as defined in claim 8, wherein said membrane is defined as a layer in a cascading style sheet web page". It is respectfully submitted that this feature is neither taught nor suggested in the cited sections of Brown, Barlow and Humes.

b. Independent Claim 18

A method for preventing data entry to a web page comprising the steps of:
associating an executable script with said web page;
permitting a first data input to said web page;
executing, in response to said first data input, said executable script; and
preventing data entry to at least a portion of said web page after execution of said script, wherein said step of preventing further comprises the steps of:
 associating said executable script with a predetermined z-index number for said web page; and
 rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number.

The Final Official Action correctly states that "Moneymaker fails to disclose associating the executable script with a predetermined z-index number for a web page and rendering inaccessible those data entry elements associated with the web page that have a z-index number lower than the predetermined z-index number." The cited

sections of Humes (Figure 5 and col. 2, lines 36-49), used in the Final Official Action to allegedly remedy the deficiency of Moneymaker et al., describe a method of filtering a page based on weighted values of the words found on the page to be accessed. However, as described above, with respect to claim 1, the cited sections of Humes do not remedy this deficiency because Humes nowhere describes the use of a z-index number at all and Humes usage of a Targetscore to filter objectionable data is not the same as the claimed "associating said executable script with a predetermined z-index number for said web page; and rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number."

However, as best understood the Final Official Action incorrectly equates the Targetscore of Humes to the z-index used in Appellant's claim 1 combination as shown by the following from the Final Official Action:

"Humes discloses associating the executable script with a predetermined z-index number for a web page and rendering inaccessible those data entry elements associated with the web page that have a z-index number lower than the predetermined z-index number (Figure 5)."

It is respectfully submitted that this analogy between the claimed Targetscore and z-index number fails for the reasons given above with respect to claim 1.

Additionally, the Moneymaker et al. reference describes methods and systems for data compilation, management and manipulation in a static interface environment. Per the abstract of Moneymaker et al. "the present invention allows a business to display all of its products for sale via a single web page and then have its customers order any of these products or their relevant subsets of options, without the customer ever having to leave this single web page." Neither the abstract of Moneymaker et al. nor the cited sections of Moneymaker et al. (paragraphs 34-39) exhibit a need or desirability for filtering objectionable data. Therefore there would have been no motivation for one of ordinary skill in the art to combine the Moneymaker et al. reference with the Humes reference.

Still further, the cited section of Humes (column 2, lines 36-49) merely describes filtering objectionable data, which is not the same as the feature found in Appellant's

claim 1. More specifically, the filtering of objectionable data of Humes prevents a user from seeing the objectionable data whereas in Appellant's claim 1 the "rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number" occurs after an end user has seen the data entry elements as shown by another feature of claim 1 "wherein said executable script operates within said client device to render said data input screen in accessible to prevent subsequent user input" (emphasis added).

c. Independent Claim 24

Independent claim 24 was rejected in the Final Official Action for reasons similar to claim 1. It is respectfully submitted that claim 24 is also similar to claim 18 and is allowable at least for the reasons described above regarding both claims 1 and 18.

Conclusions

Accordingly it is respectfully submitted that the rejection of claims 1-17 and 24 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Brown et al. and in view of Barlow et al. and further in view of Humes does not establish a *prima facie* case of obviousness and should be reversed.

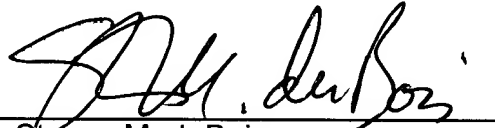
Accordingly it is respectfully submitted that the rejection of claim 18 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Moneymaker et al. in view of Humes does not establish a *prima facie* case of obviousness and should be reversed.

For at least the foregoing reasons, it is respectfully submitted that the claims are patentable over the documents cited. Accordingly, it is respectfully requested that the rejection in the Official Action of December 11, 2006 be REVERSED.

Respectfully submitted,

POTOMAC PATENT GROUP PLLC

By:



Steven M. duBois

Registration No. 35,023

Dated: July 6, 2007

(viii) **CLAIMS APPENDIX**

1. A method for preventing data entry via a data input screen on a client device, comprising:
 - rendering source code that defines said data input screen in said client device;
 - defining an executable script within said source code; and
 - executing said executable script in response to user input,wherein said executable script operates within said client device to render said data input screen inaccessible to prevent subsequent user input;
 - wherein said step of executing further comprises the steps of:
 - associating said executable script with a predetermined z-index number for a web page; and
 - rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number.
2. The method as recited in claim 1, wherein said source code comprises a tag-based language.
3. The method as recited in claim 2, wherein said source code defines a membrane layer at a higher z-index level than other Web page elements, and said step of executing said executable script further comprises changing a visibility attribute of said membrane layer.
4. The method as recited in claim 1, wherein said data input screen is received from a remote server and said step of executing said executable script is performed solely on said client device without any further processing by said remote server.
5. An apparatus for preventing entries or submissions of data via an input screen displayed on a client device, comprising:

a central processing unit;
a memory;
a user input device;
a display; and
a browser adapted to render said input screen on said display,
wherein source code is provided to said browser that contains instructions that
are interpreted by said browser to render said input screen inaccessible after an
executable script contained within said source code is executed on said client device;
wherein said source code further contains instructions which operate to:
generate association of said executable script with a predetermined z-index
number for a web page; and
render inaccessible those data entry elements associated with said web page
that have a z-index number lower than said predetermined z-index number.

6. The apparatus as defined in claim 5, wherein said executable code is
executed in response to user input.

7. The apparatus as defined in claim 5, wherein said source code is a tag-
based language.

8. The apparatus as defined in claim 5, wherein said source code defines a
membrane, and wherein a visibility attribute of said membrane is changed by said
executable script.

9. The apparatus as defined in claim 8, wherein said membrane is defined as
a layer in a cascading style sheet web page.

10. A computer-readable medium having computer-executable components
comprising:
a form definition component defining a data input screen and a data submission

field;

a style definition component defining a layer having a width and height at least as large as said data submission field;

a function definition component responsive to said data submission field, wherein upon execution of said function definition component, said layer operates to render said data submission field inaccessible on said form;

wherein said computer-executable components are operable to perform the steps of:

associating said executable script with a predetermined z-index number for a web page, and

rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number.

11. The computer-readable medium having computer-executable components as recited in claim 10, wherein said layer is initially defined as hidden, and is made visible upon execution of said function definition.

12. The computer-readable medium having computer-executable components as recited in claim 11, wherein said layer comprises one of plural layers in a cascading style sheet web page.

13. The computer-readable medium having computer-executable components as recited in claim 10, wherein said function definition component is executed in response to user operation of said data submission field.

14. The computer-readable medium having computer-executable components as recited in claim 10 wherein said function definition component is executed solely within a client device to prevent subsequent data entry via said data input screen.

15. A method for preventing data entry to a server computer from a client

computer, comprising:

- receiving a request for an exchange of data from said client computer;
- defining an executable script within a source code, said executable script operating in response to a client computer input and rendering a data input screen inaccessible to prevent subsequent input from said client computer; and
- providing said source code that defines said data input screen;

wherein said step of defining further comprises the steps of:

- associating said executable script with a predetermined z-index number for a web page; and
- rendering inaccessible those data entry elements associated with said web page that have a z-index number lower than said predetermined z-index number.

16. The method as recited in claim 15, wherein said source code comprises a tag-based language.

17. The method as recited in claim 16, wherein said source code defines a membrane layer at a higher z-index number than other Web page elements, said step of executing said executable script further comprises changing a visibility attribute of said membrane layer.

18. A method for preventing data entry to a web page comprising the steps of:

- associating an executable script with said web page;
- permitting a first data input to said web page;
- executing, in response to said first data input, said executable script; and
- preventing data entry to at least a portion of said web page after execution of said script, wherein said step of preventing further comprises the steps of:

- associating said executable script with a predetermined z-index number for said web page; and
- rendering inaccessible those data entry elements associated with said

web page that have a z-index number lower than said predetermined z-index number.

19-23. (Cancelled)

24. A method for preventing data entry to a web page comprising the steps of:
associating an executable script with said web page;
determining if said web page uses z-index numbers;
permitting a first data input to said web page;
executing, in response to said first data input, said executable script; and
preventing data entry to at least a portion of said web page after execution of
said script, wherein said step of preventing further comprises the steps of:
 associating said executable script with a predetermined z-index number
for said web page if said web page supports using said z-index number;
 associating said executable script with a division of said web page if said
web page does not support using said z-index number;
 rendering inaccessible those data entry elements associated with said
web page by rendering said division of said web page visible over said data entry
elements if said web page does not support using said z-index number; and
 rendering inaccessible those data entry elements associated with said
web page that have a z-index number lower than said predetermined z-index number if
said web page supports using said z-index number.

(ix) **EVIDENCE APPENDIX**

None.

(x) **RELATED PROCEEDINGS APPENDIX**

None.